Design Pattern: Strategy Pattern

2019.02.10.

Interface

Delegate

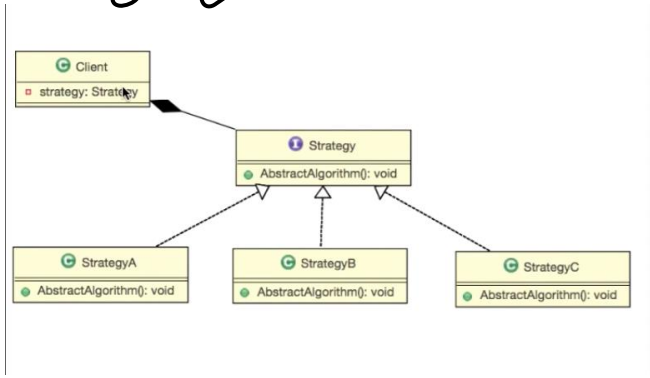
```java
public class AObj {


    private void funcAA() {
        //(훨씬 더 복잡한 기능이겠지만)
        System.out.println("AAA");
        System.out.println("AAA");
        //~기능이 필요합니다. 개발해주세요.
    }
}

```

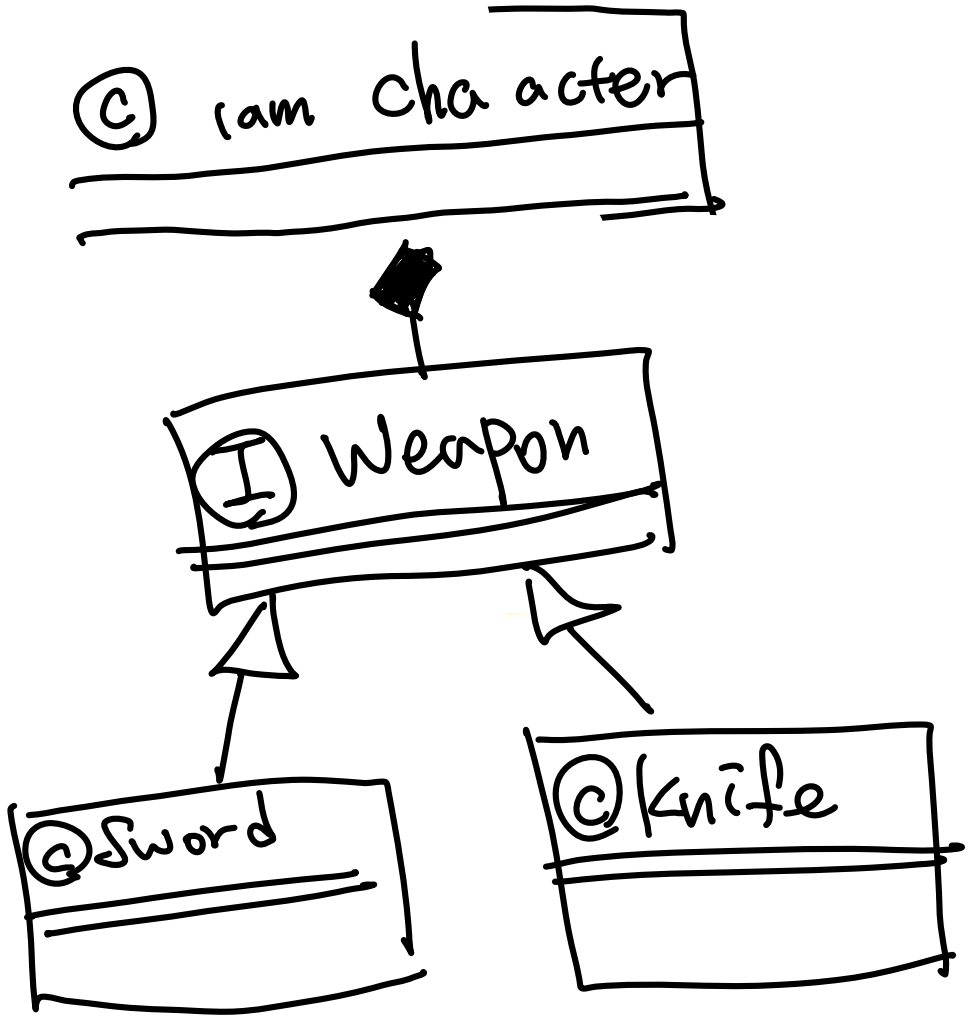
```java
public class AObj {

    Ainterface ainterface;

    private void funcAA() {
        //훨씬 더 복잡한 기능이겠지만
        ainterface.funcA();
        ainterface.funcA();
        //~기능이 필요합니다. 개발해주세요.
    }
}

```

```java
package StrategyPattern;

public class AObj {

    Ainterface ainterface;

    //Ainterface할당 for Maintest
    public AObj(){
        ainterface = new AinterfaceIMPLE();
    }
    public void funcAA() {
        //훨씬 더 복잡한 기능이겠지만
        ainterface.funcA();
        ainterface.funcA();
        //~기능이 필요합니다. 개발해주세요.
    }
}

```

Strategy Pattern

- 여러 알고리즘을 하나의 추상적인 접근점을 만들어 접근점에서 교환가능하도록 하는 패턴 (Interface)





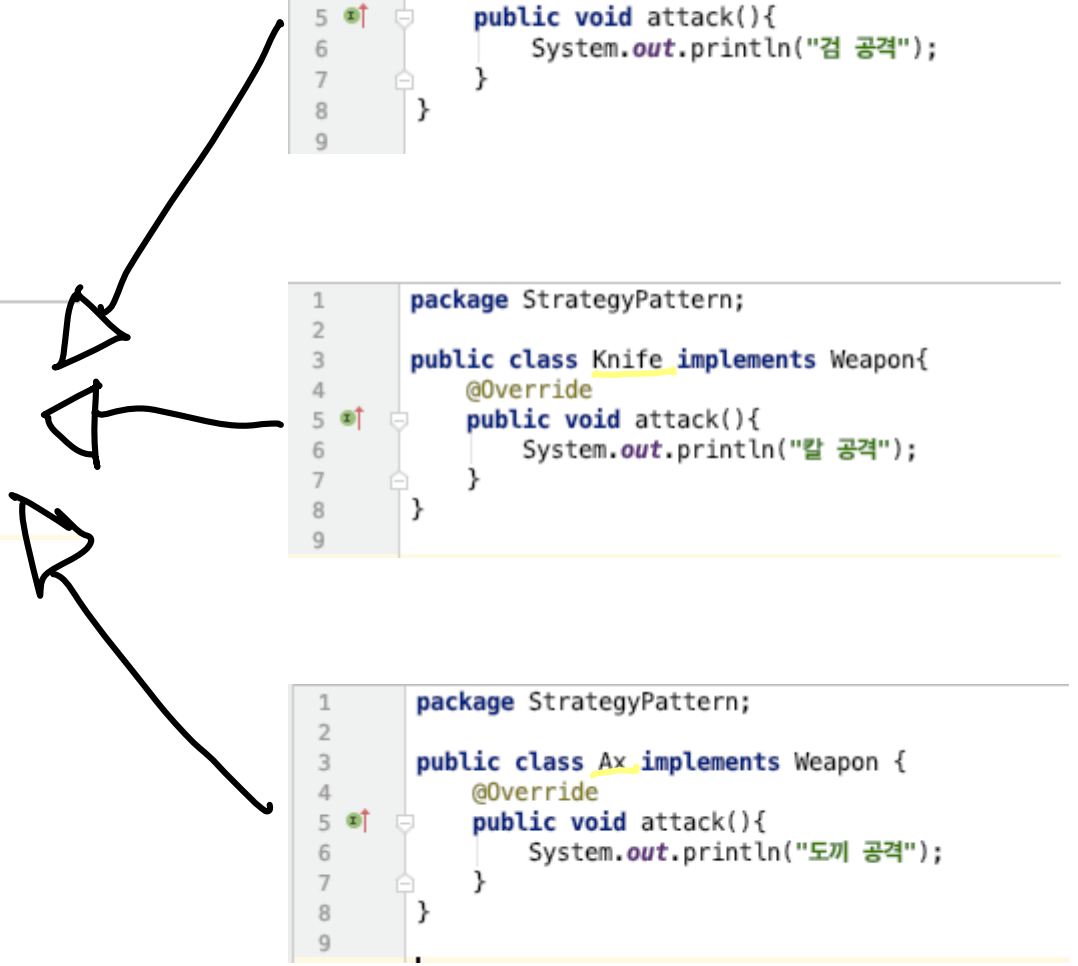
Strategy Pattern

```java
package StrategyPattern;

public interface Weapon {
    public void attack();
}
```

```java
package StrategyPattern;

public class Sword implements Weapon {
    @Override
    public void attack(){
        System.out.println("검 공격");
    }
}
```

```java
package StrategyPattern;

public class Knife implements Weapon{
    @Override
    public void attack(){
        System.out.println("칼 공격");
    }
}
```

```java
package StrategyPattern;

public class Ax implements Weapon {
    @Override
    public void attack(){
        System.out.println("도끼 공격");
    }
}
```



Strategy Pattern

알고리즘이 추가되어도 유지보수가 용이하다!

```java
package StrategyPattern;

public class GameCharacter {

    //접근점-추상적인 접근점을 만든다.
    private Weapon weapon;

    //교환가능하도록 만든다
    public void setWeapon(Weapon weapon){
        this.weapon = weapon;
    }

    //왜 만드는가? 기능을 사용하고자 만든다.
    public void attack(){
        if(weapon == null){
            System.out.println("맨손공격");
        }else{
            //델리게이트
            weapon.attack();
        }
    }
}
```

```java
package StrategyPattern;

public class Main {
    public static void main(String[]args){

        GameCharacter character = new GameCharacter();

        character.attack();

        character.setWeapon(new Knife());
        character.attack();

        character.setWeapon(new Sword());
        character.attack();

        //무기 추가하는데 굉장히 효율적이다.
        //알고리즘이 추가되어도 유지보수가 어렵지 않다.
        character.setWeapon(new Ax());
        character.attack();

    }
}
```